

# Nachdenkzettel JavaFX

## Aufgabe 1:

Ich würde alle Elemente in eine **VBox** stecken, um die Elemente in einer Spalte anordnen zu können. Die Elemente pro Zeile würde ich innerhalb der VBox in **HBox**-en anordnen. Damit z.B. das Firstname-Label neben dem entsprechenden Textfeld ist. Den Abstand könnte man hier dann mit dem spacing-Attribut einstellen. In der letzten Zeile mit den drei Buttons kann man dann noch per `hbox4.setAlignment(Pos.CENTER)`; die gesamte 4. HBox zentrieren.

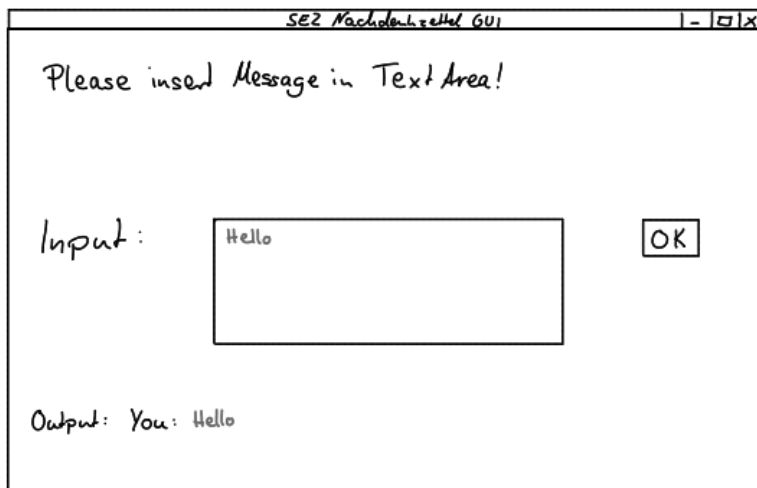
## Aufgabe 2:

- Alter Eventhandler (anonyme, innere Klasse):  

```
Button b1 = new Button („Please push me!");  
btn1.setOnAction(new EventHandler<ActionEvent> () {  
    @Override  
    public void handle(ActionEvent event) {  
        btn1.setText("Yesss, thanks!");  
    }  
});
```
- Neuer Eventhandler (mit Lambda):  

```
Button btn2 = new Button ("Don't push me!");  
btn2.setOnAction(event -> btn2.setText("Hey! Why did you push me?!"));
```

## Aufgabe 3:



*Beispiel*

Was macht der Eventhandler?

Sobald der OK-Button angeklickt wird, nimmt der Eventhandler den eingetippten Text aus der inputArea und hängt ihn hinter „You: “ an die outputArea und ergänzt noch einen Zeilenumbruch.

## Aufgabe 4:

Dies ist keine Exception, da die Datenbankabfrage zwar kein Ergebnis bzw. ein leeres ResultSet beim Abfragen der Kunden-ID liefert, dies jedoch nicht automatisch eine Exception wirft. Je nach Weiterverarbeitung des ResultSets kann es u.U. zu einer Exception kommen, wenn z.B. ein bestimmter Index des leeren ResultSets aufgerufen werden soll. Hier kann es dann sinnvoll sein eine

eigene Exception zu werfen, um in der aufrufenden Methode dem User mitzuteilen, dass er sich noch registrieren muss.

#### **Aufgabe 4:**

Bei langlaufenden Operationen muss man beachten, dass der Eventhandler u.U. keine Rückmeldung gibt, dass gerade ein Prozess läuft. Am besten fügt man einen Lade-Screen ein, so dass der Nutzer erkennen kann, dass er warten muss und nicht weitere Eventhandler parallel aktiviert oder sogar den selben erneut startet.